Getting started with ggplot2

Data Visualization

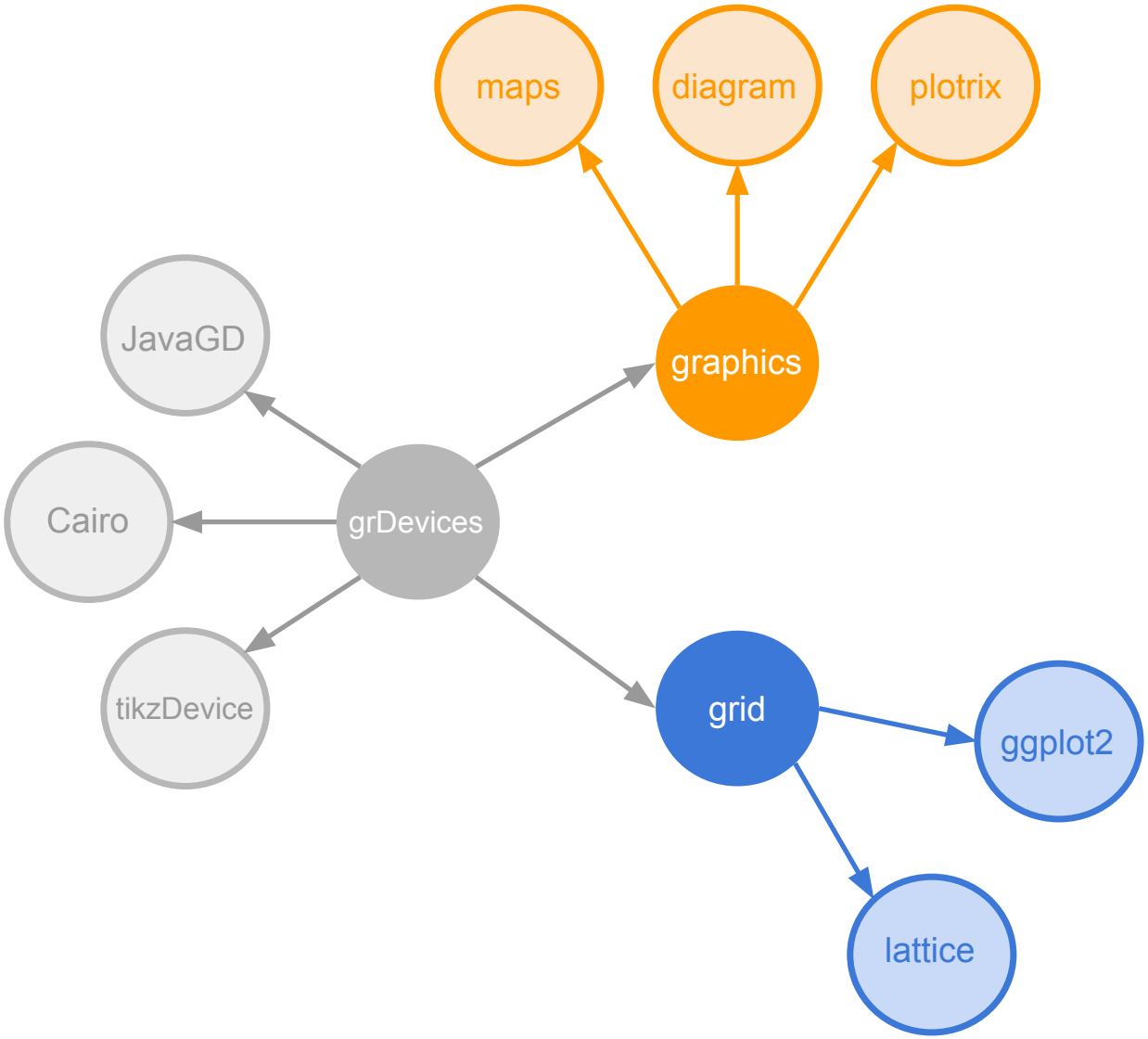
Gaston Sanchez

`github.com/ucb-stat243/stat243-fall-2016`

ggplot2

Recap

- There are two main graphic systems in R:
  - The R package `"graphics"`
  - The R package `"grid"`
- `"graphics"` is the *traditional* system
- `"grid"`
  - prodives low-level functions for programming plotting functions
  - does not provide functions for drawing complete plots.
  - is used to build other graphics packages like `"ggplot2"`.


maps
diagram
plotrix
JavaGD
graphics
Cairo
grDevices
tikzDevice
grid
ggplot2
lattice

Resources for "ggplot2"

- **Documentation**

  http://docs.ggplot2.org/

- **Cheat-sheet**

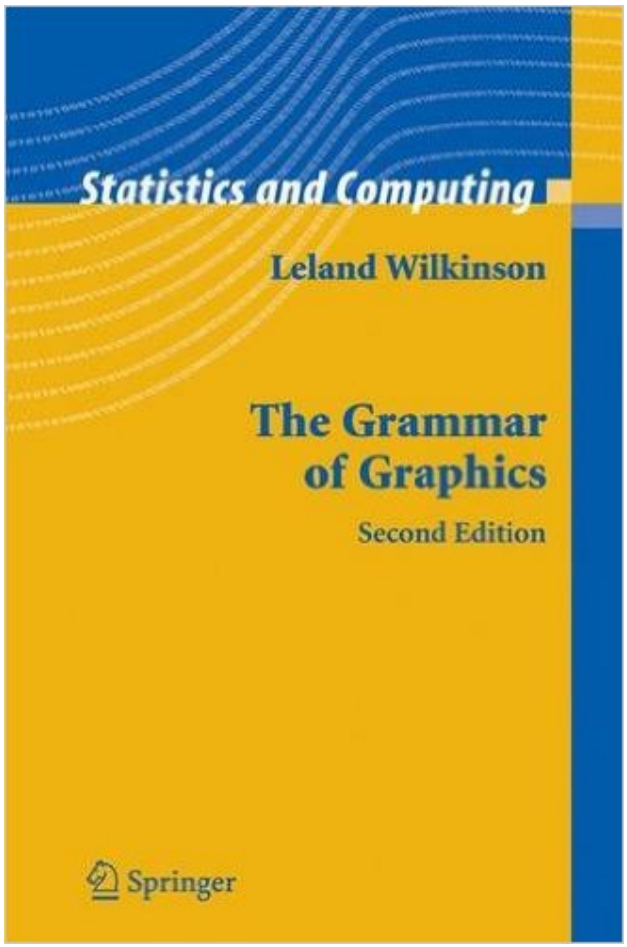
  https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

- **ggplot2: Elegant Graphics for Data Analysis** (by Hadley Wickham)
- **R Graphics Cookbook** (by Winston Chang)
- **R Graphics** (by Paul Murrell)

The Grammar of Graphics and "ggplot2"

- "ggplot2" (by Hadley Wickham) is an R package for producing statistical graphics
- It provides a framework based on Leland Wilkinson's **Grammar of Graphics**

Le Wilkinson's Grammar of Graphics



About Le Wilkinson

- 1980s teaching a seminar in statistical graphics
- wrote the SYSTAT package in the late 1980s
- President of SYSTAT Inc. 1984-1994 (bought by SPSS)
- Vice-President of SPSS
- co-wrote GPL (Graphics Programming Language) in Java

Review of GG by Nicholas Cox

- GG is Wilkinson's magnum opus
- Fruit of 30+ years of experience
- Several sections appear too enigmatic
- Under the hood, the formal notation correspond to GPL
- The meaning of the GG remains elusive
- Chapters seem to be arbitrarily organized
- co-authors: Graham Wills, Dan Rope, Andrew Norton, Rogger Dubbs

file:///Users/gaston/Downloads/v17b03.pdf

About the Grammar of Graphics

- *The Grammar of Graphics* is Wilkinson's attempt to define a theoretical framework for graphics
- **Grammar**: Formal system of rules for generating graphics
  - Some rules are mathematic
  - Some rules are aesthetic

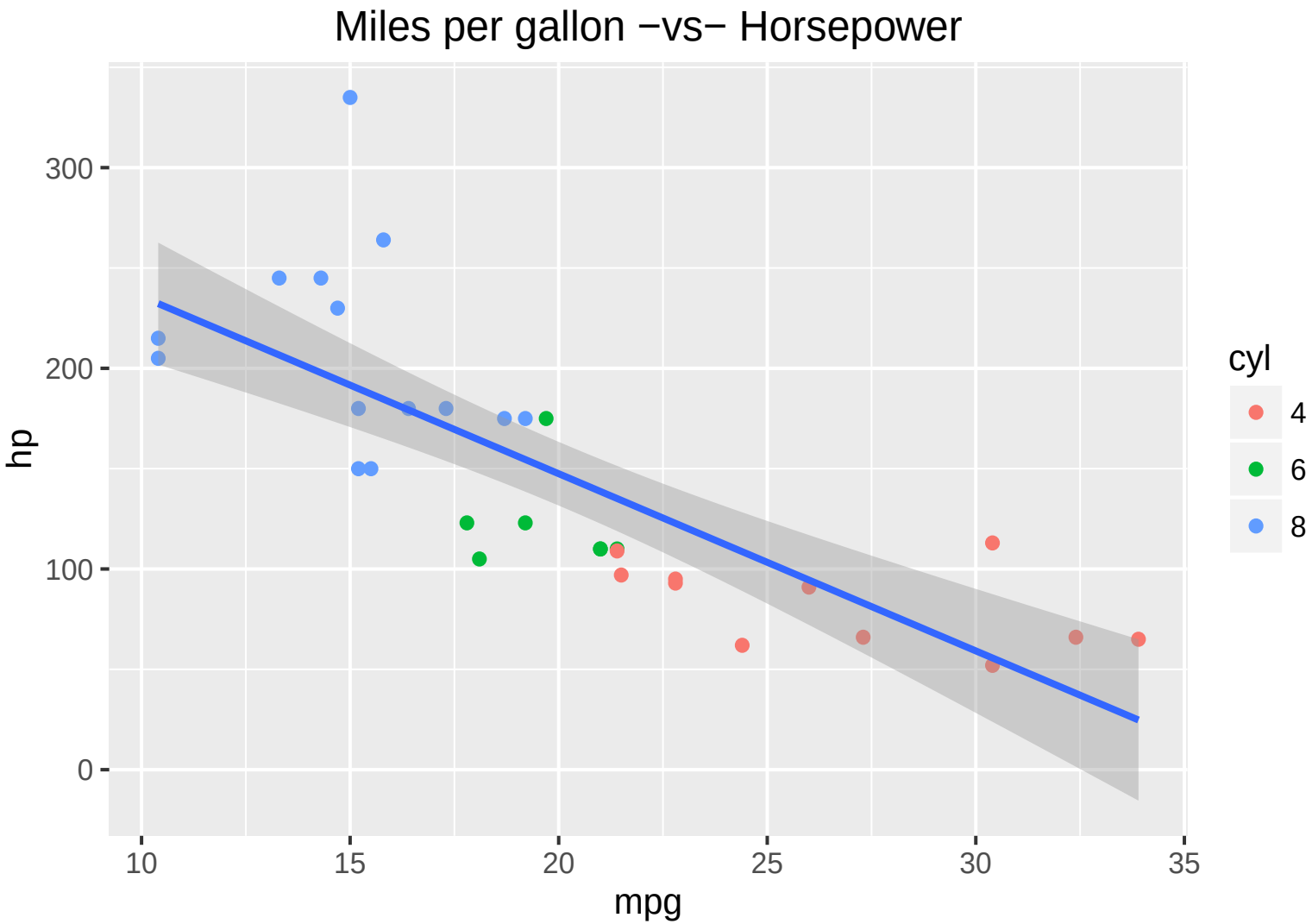
What is a Statistical Graphic?

Some Data set

```
mtcars
```

```
##                    mpg   hp  cyl
## Mazda RX4         21.0  110    6
## Mazda RX4 Wag     21.0  110    6
## Datsun 710        22.8   93    4
## Hornet 4 Drive    21.4  110    6
## Hornet Sportabout 18.7  175    8
## Valiant           18.1  105    6
## Duster 360        14.3  245    8
## Merc 240D         24.4   62    4
## Merc 230          22.8   95    4
## Merc 280          19.2  123    6
```

What is a statistical graphic?



What is a statistical graphic?

Elements to draw the chart "manually"

- coordinate system
- x and y axis (intervals)
- axis tick marks
- axis labels, and title
- points (with colors)
- regression line (and ribbon)
- legend

About the Grammar of Graphics

3 Stages of Graphic Creation

- **Specification**: link data to graphic objects
- **Assembly**: put everything together
- **Display**: render of a graphic

About the Grammar of Graphics

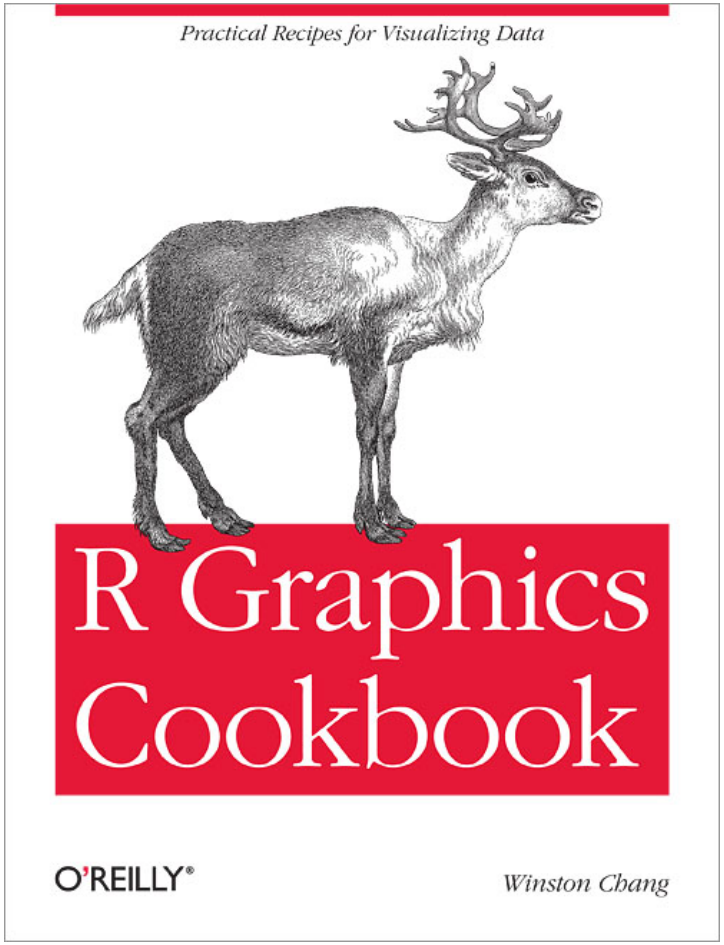
Specification

Link data to graphic objects

- Data
- Transformation of variables (e.g. aggregation)
- Scale transformations (e.g. log)
- Coordinate system (e.g. cartesian)
- Graphic Elements (e.g. points, lines)
- Guides (e.g. labels, legends)

About ggplot2

References for ggplot2



About "ggplot2"

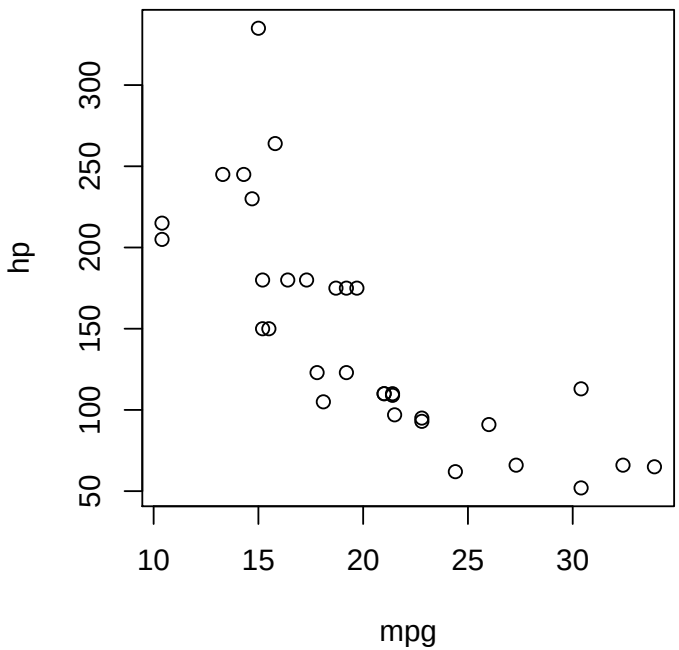
- Inspired in the **Grammar of Graphics** by Lee Wilkinson
- Developed by Hadley Wickham
- Started in early 2000s as part of Wickham's PhD
- Implementation in R as a *layered grammar of graphics*

About "ggplot2"

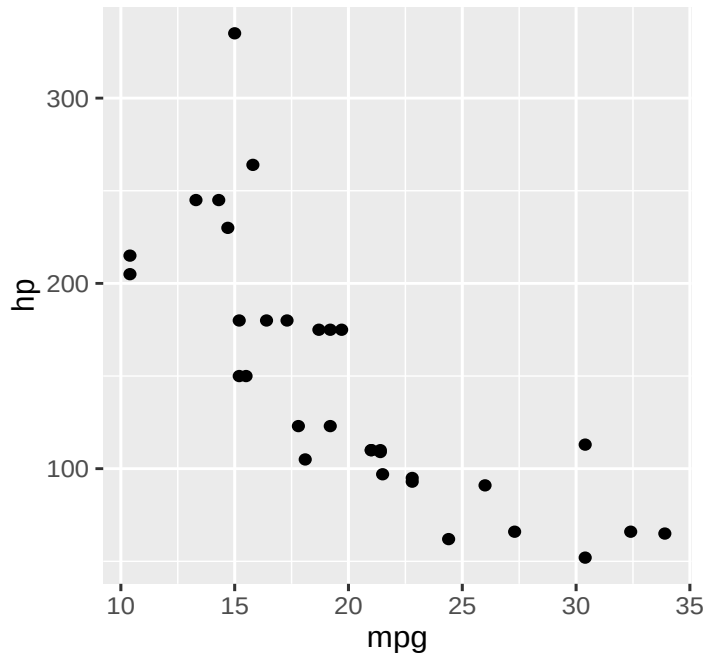
- "ggplot2" is the name of the package
- The gg in "ggplot2" stands for *Grammar of Graphics*
- "ggplot" is the class of objects (plots)
- ggplot() is the main function in "ggplot2"

Base graphics -vs- "ggplot2"

base graphics



ggplot2

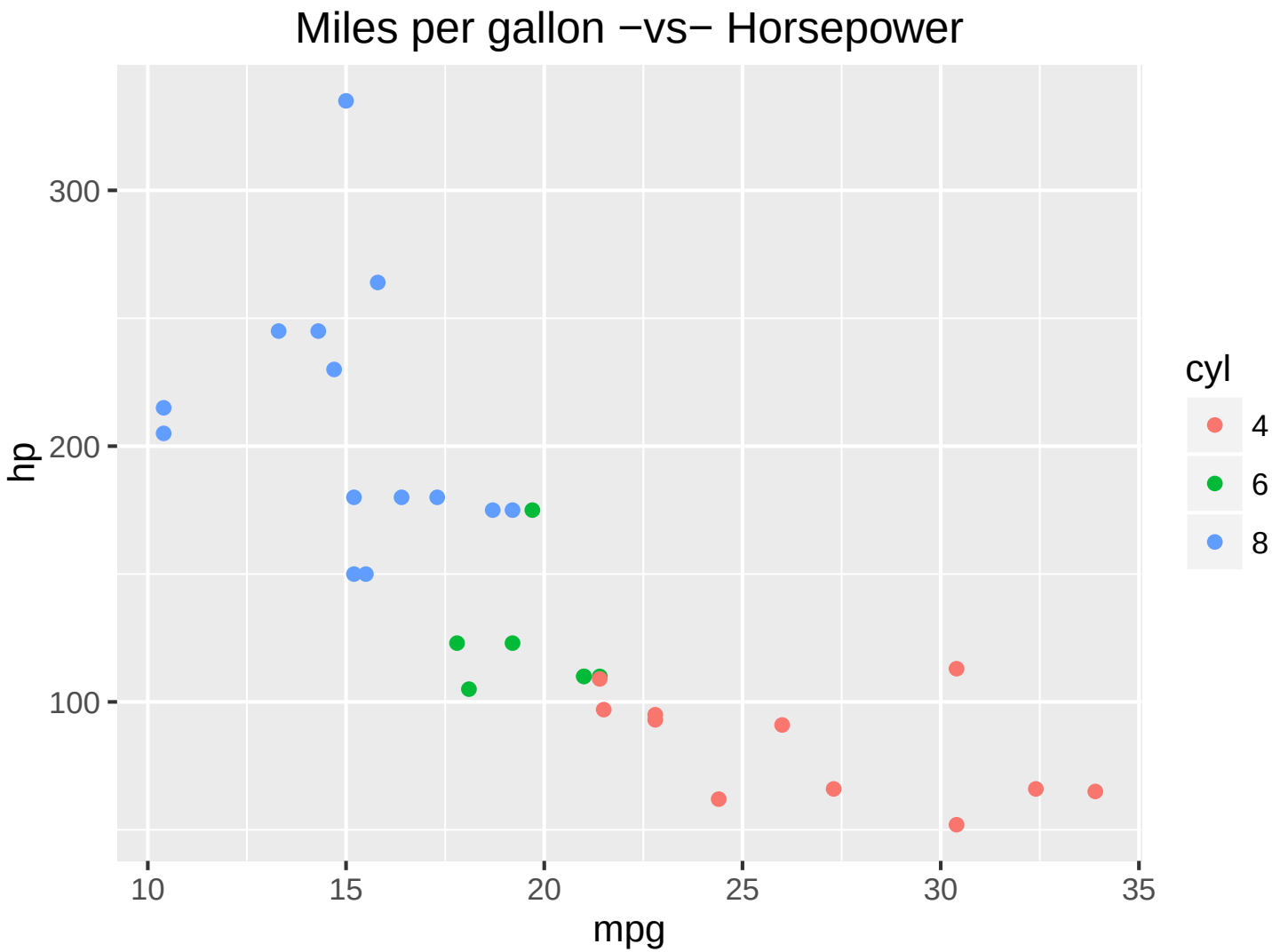


About "ggplot2"

- "ggplot2" provides beautiful plots while taking care of fiddly details like legends, axes, colors, etc.
- "ggplot2" is built on the R graphics package "grid"
- Underlying philosophy is to describe a wide range of graphics with a compact syntax and independent components

About "ggplot2" (cont'd)

- Default appearance of plots carefully chosen
- Designed with visual perception in mind
- Inclusion of some components, like legends, are automated
- Great flexibility for annotating, editing, and embedding output


Miles per gallon −vs− Horsepower
hp
300
200
100
10
15
20
25
30
35
mpg
cyl
4
6
8

Starting with "ggplot2"

package "ggplot2"

```
# remember to install ggplot2
# (just once)
install.packages("ggplot2")


# load ggplot2
library(ggplot2)


# see basic documentation
?ggplot
```

Data set mtcars

mtcars

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Data set `mtcars`

- `mpg`: fuel efficienty in miles per gallon
- `cyl`: number of cylinder
- `disp`: size of the car engine (displacement)
- `hp`: gross horsepower
- `drat`: rear axle ratio
- `wt`: weight (1000 lbs)
- `qsec`: 1/4 mile time
- `vs`: V/S
- `am`: transmission (0 = automatic, 1 = manual)
- `gear`: number of forward gears
- `carb`: number of carburetors

mtcars subset

Consider variables mpg, hp, and cyl (as factor)

```
##                    mpg   hp  cyl
## Mazda RX4         21.0  110    6
## Mazda RX4 Wag     21.0  110    6
## Datsun 710        22.8   93    4
## Hornet 4 Drive    21.4  110    6
## Hornet Sportabout 18.7  175    8
## Valiant           18.1  105    6
## Duster 360        14.3  245    8
## Merc 240D         24.4   62    4
## Merc 230          22.8   95    4
## Merc 280          19.2  123    6
```

Making a ggplot

- The first step involves specifying a data set containing the variables to be visualized.
- the data set must be in a data frame

```
obj <- ggplot(data = mtcars)
```

`obj` is an object of class `"ggplot"`

Geoms and Aesthetics

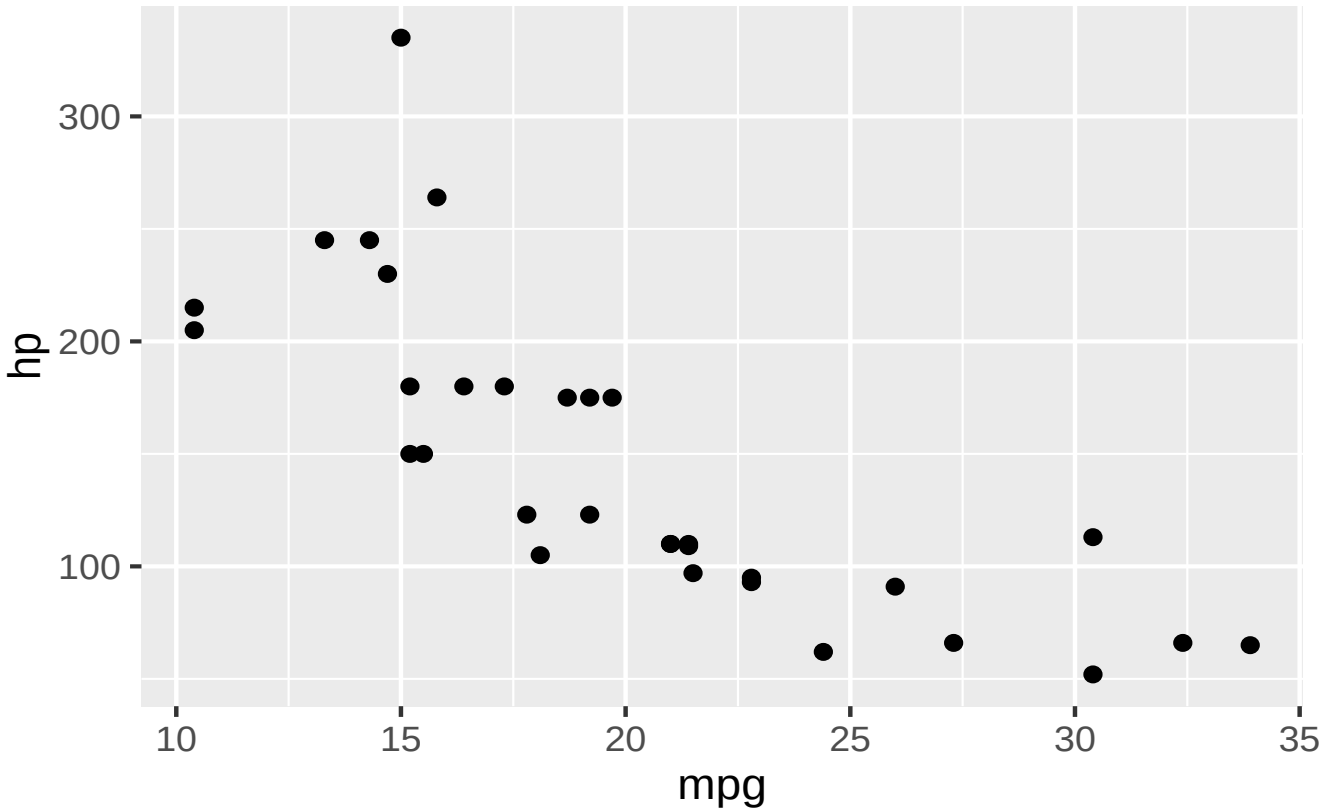
- The second step involves specifying what sort of geometric object will be used
- you also need to specify which variables will be used to control the features of the *geoms*

```
obj + geom_point(aes(x = mpg, y = hp))
```

Each geom has its associated aesthetic attributes

Scatterplot with "ggplot2"

```
obj + geom_point(aes(x = mpg, y = hp))
```



Geoms and Aesthetics

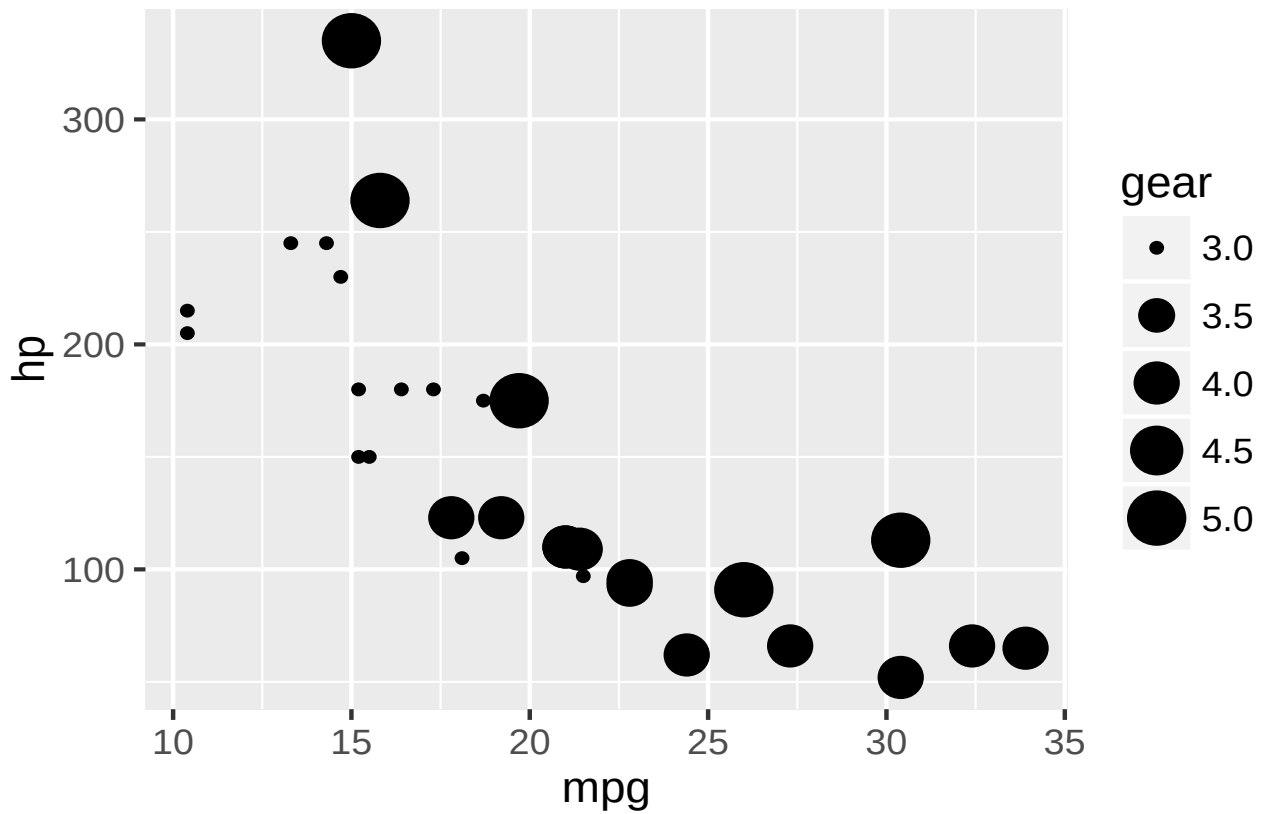
- Aesthetic attributes can be either *mapped* or *set*
- when you use a variable you *map* an aesthetic
- when you use a fixed value you *set* an aesthetic

```
# mapping size to gear
obj + geom_point(aes(x = mpg, y = hp, size = gear))

# setting size to 3
obj + geom_point(aes(x = mpg, y = hp), size = 3)
```

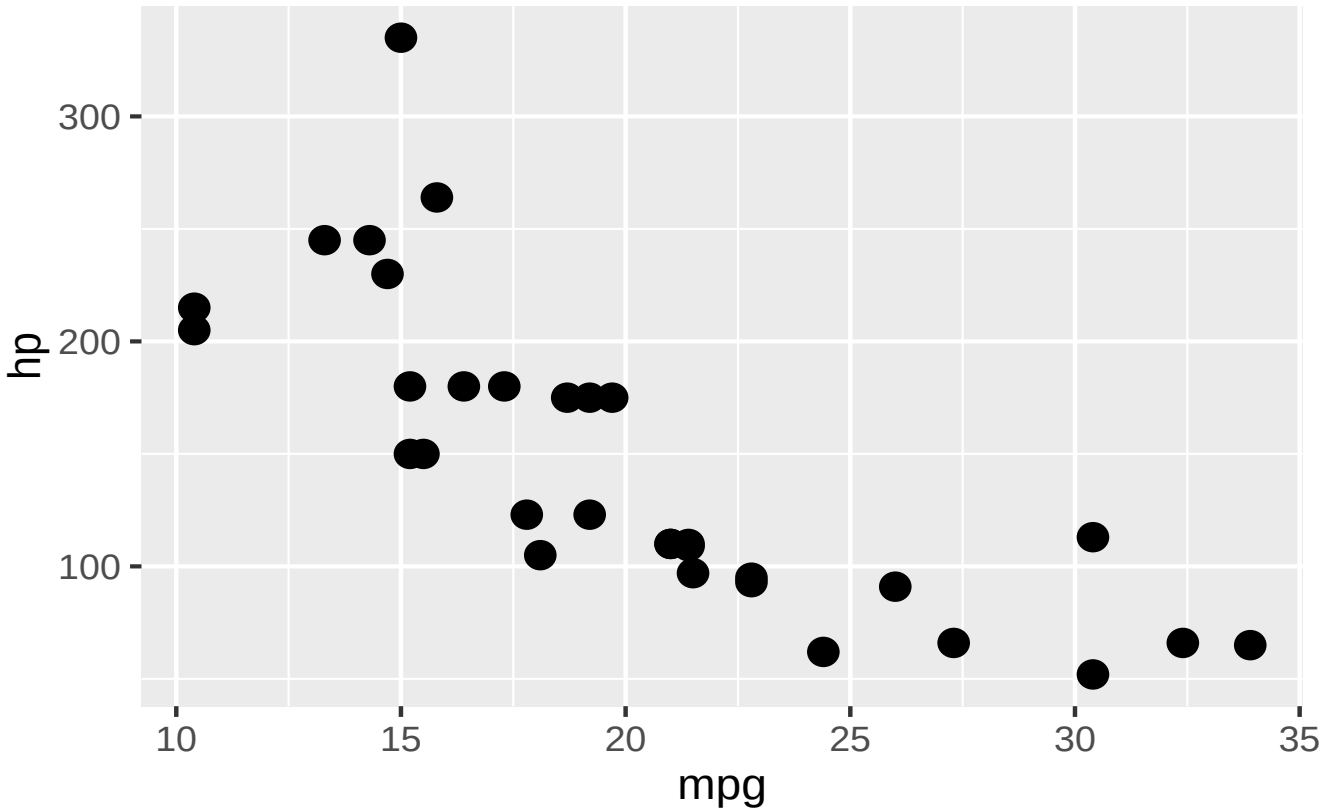
Mapping an aesthetic attribute

```
obj + geom_point(aes(x = mpg, y = hp, size = gear))
```



Setting an aesthetic attribute

```
obj + geom_point(aes(x = mpg, y = hp), size = 3)
```



Steps in creating a plot with `ggplot2`

- specify the data that you want to plot and create an empty plot object with `ggplot()`
- specify the graphics shapes or **geoms** to be used (e.g. data symbols of lines)
- specify which features or **aesthetics** of the geoms will be used to represent the data values

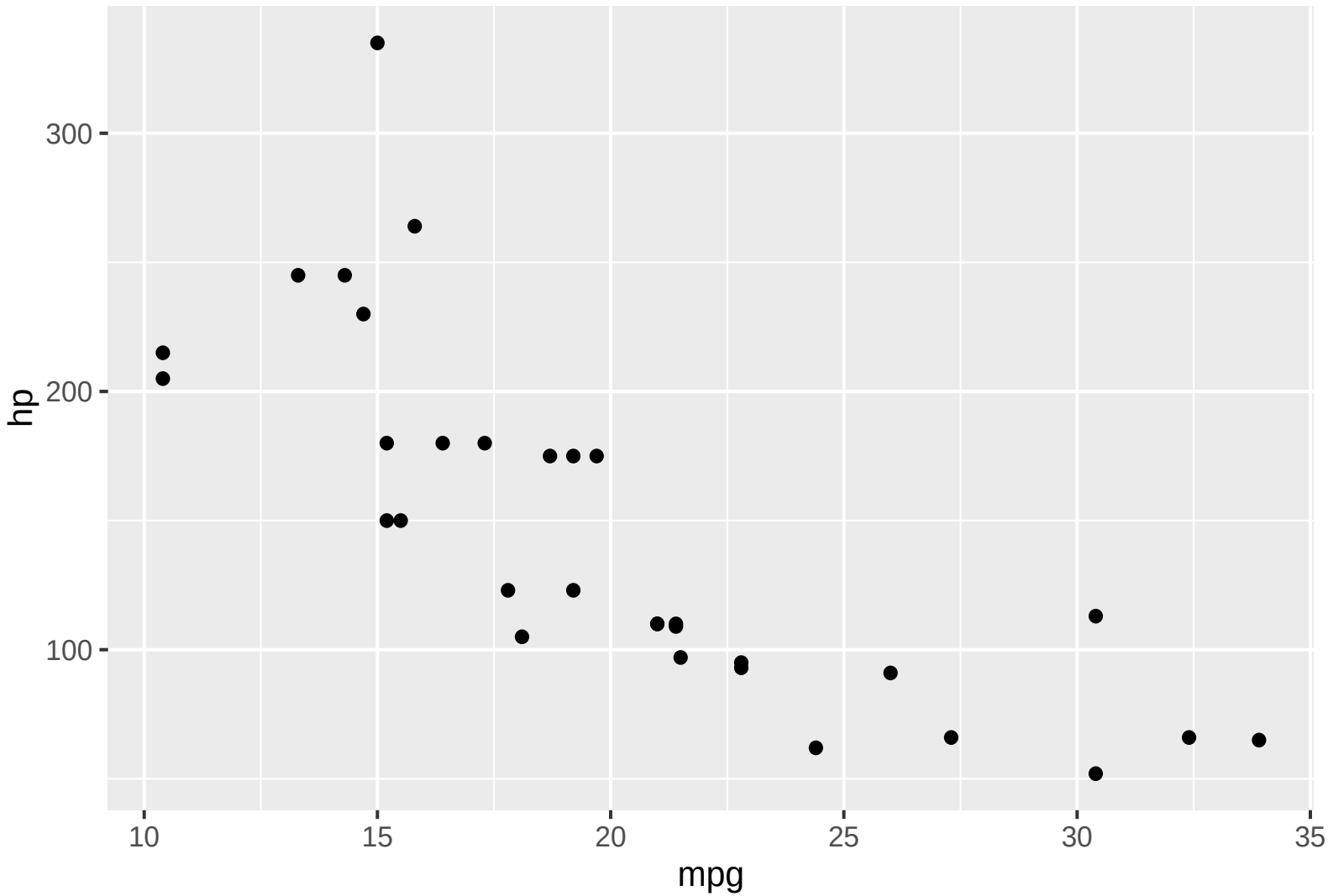
Terminology

- **data** consists of *variables*, which are stored in a data frame
- **geoms** are the geometric objects that are drawn to represent the data (e.g. bars, lines, points)
- **aesthetic attributes** are visual properties of geoms, such as position, line color, point shape
- **mapping** is a correspondance from data values to aesthetics
- **scales** control the mapping from the values in the data space to values in the aesthetic space
- **guides** show the viewer how to map the visual properties back to the data space.

Basic scatterplot

```
ggplot(data = mtcars) +
  geom_point(aes(x = mpg, y = hp))
```

Basic scatterplot



Steps in creating a plot with `ggplot2`

- A plot is built up by creating plot components or **layers** and combining them using the + operator
- specify the graphics shapes or **geoms** to be used (e.g. data symbols of lines)
- mapping or setting visual (aesthetic) attributes

Scatterplot with "ggplot2"

- `ggplot()` initializes a "ggplot" object
- specify the dataset with `data`
- type of geometric object: `geom_point()`
- mapping aesthetic attributes to variables with `aes()`
  - x-position: `mpg`
  - y-position: `hp`
  - color: `cyl`

Scatterplot with "ggplot2"

Automated things in "ggplot2"

- Axis labels
- Legends (position, labels, symbols)
- Choose of colors for points
- Background color (e.g. gray)
- Grid lines (major and minor)
- Axis tick marks

you can always change the automated elements

"ggplot2" graphics

Philosophy of "ggplot2"

A graphic is a **mapping** from **data** to **aesthetic attributes** (color, shape, size) of **geometric objects** (points, lines, bars)

- `ggplot(data, ...)`
- `aes()`
- `geom_objects()`

Scatterplot with "ggplot2"

How does "ggplot2" work?

- plots are created piece-by-piece
- plot components added with + operator
- aesthetic attributes mapped or set to data values
- computation of scales for aesthetic attributes

How does it work?

Usually, we specify the data and variables inside the function `ggplot()`

```
ggplot(data = mtcars, aes(x = mpg, y = hp))
```

Note the use of the internal function `aes()` to *map* `x` to `mpg`, and `y` to `hp`.

Then we **add a layer** of geometric objects: points in this case

```
+ geom_point()
```

Some alternative options

```
# option A
ggplot(data = mtcars,
       aes(x = mpg, y = hp, color = cyl)) +
  geom_point()
```

Some alternative options

```
# option A
ggplot(data = mtcars,
       aes(x = mpg, y = hp, color = cyl)) +
  geom_point()
```

```
# option B
ggplot(data = mtcars) +
  geom_point(aes(x = mpg, y = hp, color = cyl))
```

Some alternative options

```
# option A
ggplot(data = mtcars,
       aes(x = mpg, y = hp, color = cyl)) +
  geom_point()
```

```
# option B
ggplot(data = mtcars) +
  geom_point(aes(x = mpg, y = hp, color = cyl))
```

```
# option C
ggplot() +
  geom_point(data = mtcars,
             aes(x = mpg, y = hp, color = cyl))
```

Main inquiries

Always ask yourself ...

- What is the **data** set of interest?
- What **variables** will be used to make the plot?
- What **graphics shapes** will be used to display?
- What **features** of the shapes will be used to represent the data values?

"ggplot2" basics

- The data must be in a `data.frame`
- Variables are mapped to aesthetic attributes
- Aesthetic attributes belong to geometric objects **geoms** (points, lines, polygons)

Basic Terminology

- **ggplot()** - The main function where you specify the dataset and variables to plot
- **geoms** - geometric objetcs
  - `geom_point()`, `geom_bar()`, `geom_line()`, `geom_density()`
- **aes** - aesthetics (i.e. attributes)
  - shape, color, fill, linetype

Warning

"ggplot2" comes with the function `qplot()` (i.e. *quick plot*). Avoid using it!

As Karthik Ram says: "you'll end up unlearning and relearning a good bit"